Linearizing the Method of Conjugate Gradients

Gratton, Serge; Titley-Peloquin, David; Toint, Philippe; Tshimanga Ilunga, Jean

Published in:
SIAM Journal on Matrix Analysis and Applications

DOI:
10.1137/120889848

Publication date:
2014

Document Version
Peer reviewed version

Linearizing the Method of Conjugate Gradients

by S. Gratton[1,3], D. Titley-Peloquin,[3], Ph. L. Toint[2] and J. Tshimanga[1]

Report NAXYS-15-2012 3 September 2012

[1] ENSEEIHT, 2, rue Charles Camichel, 31000 Toulouse (France)

[2] University of Namur, 61, rue de Bruxelles, B5000 Namur (Belgium)

[3] CERFACS, 42, avenue Gaspard Coriolis, 31057 Toulouse (France)

http://www.fundp.ac.be/sciences/naxys

LINEARIZING THE METHOD OF CONJUGATE GRADIENTS*

SERGE GRATTON†, DAVID TITLEY-PELOQUIN‡, PHILIPPE TOINT§, AND JEAN TSHIMANGA ILUNGA¶

Abstract. The method of conjugate gradients (CG) is widely used for the iterative solution of large sparse systems of equations $Ax = b$, where $A \in \Re^{n\times n}$ is symmetric positive definite. Let x_k denote the k–th iterate of CG. In this paper we obtain an expression for J_k, the Jacobian matrix of x_k with respect to b. We use this expression to obtain computable bounds on the spectral norm condition number of x_k, and to design algorithms to compute or estimate $J_k v$ and $J_k^T v$ for a given vector v. We also discuss several applications in which these ideas may be used. Numerical experiments are performed to illustrate the theory.

Key words. Conjugate Gradients Algorithm, Lanczos Algorithm, Perturbation Analysis, Linearization, Automatic Differentiation

AMS subject classifications. 65F10, 65F20, 65F22, 65F35

1. Introduction. The method of conjugate gradients (CG) of Hestenes and Stiefel [12] is widely used for the iterative solution of large sparse systems of equations $Ax = b$, where $A \in \Re^{n\times n}$ is symmetric positive definite. Let x_k denote the k–th iterate of CG. This iterate is a nonlinear differentiable function of b. In this paper we obtain an expression for the Jacobian of x_k with respect to the right-hand side vector b,

$$J_k = \frac{\partial x_k}{\partial b} \in \Re^{n\times n}.$$

The norm of the Jacobian (for any chosen norm) is by definition the absolute condition number of x_k with respect to perturbations in b. We use our expression for the Jacobian to obtain computable bounds on the spectral norm condition number of x_k. We also discuss methods to compute or estimate the quantities $J_k v$ and $J_k^T v$ for a given vector v.

Sensitivity analysis of computational problems is considered by many to be a very important topic. Reference books in numerical linear algebra (such as [2], [6], [13]) typically present a sensitivity analysis of the problems they consider. For example, for the exact solution $x = A^{-1}b$, the Jacobian is

$$\frac{\partial(A^{-1}b)}{\partial b} = A^{-1}$$

and the condition number of x in the spectral norm is $\|A^{-1}\|_2$, the reciprocal of the smallest singular value of A. Hence we are warned that the exact solution $x = A^{-1}b$

*The research presented in this paper was conducted with the support of the "Assimilation de Données pour la Terre, l'Atmosphère et l'Océan (ADTAO)" project, funded by the "Fondation Sciences et Technologies pour l'Aéronautique et l'Espace (STAE)", Toulouse, France, within the "Réseau Thématique de Recherche Avancée (RTRA)".

†IRIT-CERFACS, 42 ave Gaspard Coriolis, 31057 Toulouse, France (gratton@cerfacs.fr).

‡IRIT-ENSEEIHT, 2 rue Charles Camichel, B. P. 7122, 31071 Toulouse, France (dtitleyp@enseeiht.fr). The research of this author was supported by a postdoctoral fellowship from Natural Sciences and Engineering Research Council (NSERC) of Canada.

§NAXYS, University of Namur, 61 rue de Bruxelles, B5000 Namur, Belgium (philippe.toint@fundp.ac.be).

¶Serviware, Technoparc 1 - Bâtiment 4 - Voie 5 L'Occitane, 31670 Toulouse, France (jtshimanga@serviware.com).

may be very sensitive to perturbations in b (such as noise in the data or rounding errors in finite precision arithmetic) when A has some very small singular values.

In many practical applications, however, one is not concerned with the exact solution $x = A^{-1}b$. Rather, one performs $k \ll n$ iterations of CG and accepts the resulting iterate as the computed "solution" even though it may be very far from $A^{-1}b$. For instance, when n is very large, this may be the only feasible approach. Performing $k \ll n$ steps of CG is also considered a form of regularization when A is very ill-conditioned; see for example the survey in [11]. Then one may wonder at quantifying this effect as a function of b. This is the question we consider in the present manuscript.

There has been some work done on the sensitivity of Krylov subspace methods. Kuznetsov et. al. [1, 16] obtain expressions for the condition number of a Krylov subspace $\mathcal{K}_k(A, b)$ with respect to perturbations in A and b. Here, however, we are interested in the sensitivity not of a whole subspace but of only one vector in the space, namely, x_k. We also mention the papers of Greenbaum [8] and Strakoš [26] (see also [10]) who consider the sensitivity of CG iterates to changes in the eigenvalue distribution of A. A summary and more thorough bibliographies can be found in [19, 20]. One important aim of such work is to understand how rounding errors in finite precision arithmetic affect the convergence of the algorithm. Here our motivation is different: we are interested in applications in which b is an observation vector greatly contaminated by noise, and in which very few iterations of CG are performed. In such cases, a sensitivity analysis of x_k with respect to perturbations in b is certainly relevant.

One application of this work is in statistics. Consider the linear model $Kx = c$, where K has full column rank. It is well known (see, e.g., [14, §17]) that given an observation $\tilde{c} = c + v$ with $v \sim (0, \Sigma)$, Σ non-singular, the least-squares estimator

$$x_{LS} = \arg\min_x \|\tilde{c} - Kx\|_{\Sigma^{-1}} = (K^T\Sigma^{-1}K)^{-1}K^T\Sigma^{-1}\tilde{c}$$

is the best linear unbiased estimator of x, and

$$\mathrm{cov}\{x_{LS} - x\} = (K^T\Sigma^{-1}K)^{-1}.$$

But if x_k is the k-th iterate of CG applied to the normal equations of the above least-squares problem, what is the covariance of $x_k - x$, and how does it compare to that of $x_{LS} - x$? This is sometimes called the partial least-squares problem in the statistics literature; see for example [3] and the references therein. If we know the Jacobian of x_k, we have to first order

$$\mathrm{cov}\{x_k - x\} \approx J_k\Sigma J_k^T.$$

See e.g. [5, 21] and the references therein for some data assimilation applications in which knowledge of $\mathrm{cov}\{x_k - x\}$ would be useful.

Another potential application is to solving linear systems with multiple right-hand sides $Ax = b_i$, $i = 1, 2, \ldots$. Suppose that after k steps of CG applied to $Ax = b_i$ we have computed $x_k(b_i)$ and $J_k(b_i)$. To first order

$$x_k(b_{i+1}) \approx x_k(b_i) + J_k(b_i) \cdot (b_{i+1} - b_i),$$

which is an ideal starting vector for CG applied to $Ax = b_{i+1}$. If $J_k \approx A^{-1}$, the Jacobian can be used as a preconditioner for subsequent solves with A and to estimate the energy norm of the error,

$$\|\epsilon_k\|_A^2 = \epsilon_k^T A\epsilon_k = r_k^T A^{-1} r_k \approx r_k^T J_k r_k,$$

where $\epsilon_k = A^{-1}b - x_k$ is the error and $r_k = b - Ax_k$ the residual. In each of the above applications, in practise we would more likely use an approximation to J_k with which matrix-vector multiplications can be performed cheaply, rather than J_k itself. We discuss a few such approximations.

The rest of this paper is organized as follows. In Section 2 we introduce the Lanczos [17] and CG algorithms. In Section 3 we obtain an expression for J_k, the Jacobian of x_k with respect to b. We also give bounds on the normwise relative error between J_k and A^{-1} and on $\|J_k\|_2$. We discuss methods to compute or estimate $J_k v$ and $J_k^T v$ for a given vector v in Section 4. In Section 5 we present numerical experiments to illustrate the theory, and we conclude with a discussion in Section 6.

2. The Lanczos and Conjugate Gradients algorithms. We start by reviewing some known facts about the Lanczos algorithm and its relation to CG. We assume exact arithmetic. The effects of rounding errors in floating point arithmetic are discussed briefly in Section 6. For a more thorough treatment of these topics, including implementation details, see for example [2], [4], [9], [19].

The Lanczos algorithm computes an orthogonal tridiagonalization of the symmetric matrix $A \in \Re^{n\times n}$ column by column, starting from an arbitrary normalized vector v_1. After k steps the algorithm produces $V_k = \begin{bmatrix} v_1, \dots v_k \end{bmatrix} \in \Re^{n\times k}$ with orthonormal columns and

$$T_k = \begin{bmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_k \\ & & \beta_k & \alpha_k \end{bmatrix}, \qquad \widetilde{T}_k = \begin{bmatrix} T_k \\ \beta_{k+1} e_k^T \end{bmatrix}, \tag{2.1}$$

such that

$$AV_k = V_k T_k + \beta_{k+1} v_{k+1} e_k^T = V_{k+1} \widetilde{T}_k. \tag{2.2}$$

(Here e_k is the k–th standard basis vector, not to be confused with the error ϵ_k below.) The columns of V_k form an orthonormal basis for the Krylov subspace

$$\mathcal{K}_k(A, v_1) = \text{span}\{v_1, Av_1, \dots, A^{k-1}v_1\}.$$

Starting from an initial guess x_0 with residual $r_0 = b - Ax_0$, CG produces a sequence of iterates satisfying

$$\begin{aligned} &x_k \in x_0 + \mathcal{K}_k(A, r_0), \\ &r_k = b - Ax_k \in A\mathcal{K}_k(A, r_0), \qquad r_k \perp \mathcal{K}_k(A, r_0). \end{aligned}$$

Define $\beta_1 = \|r_0\|_2$ and start the Lanczos algorithm with $v_1 = r_0/\beta_1$. Then the iterate x_k and residual r_k from CG can be written as

$$\begin{aligned} x_k &= x_0 + V_k T_k^{-1} \beta_1 e_1 \\ r_k &= r_0 - AV_k T_k^{-1} \beta_1 e_1 = -\beta_1 \beta_{k+1} e_k^T T_k^{-1} e_1 v_{k+1}. \end{aligned} \tag{2.3}$$

We will also use the fact that $\mathcal{K}_k(A, r_0) = \text{Range}(V_k) = \text{Range}(K_k)$, where

$$K_k = \begin{bmatrix} r_0, \dots, A^{k-1} r_0 \end{bmatrix} = \begin{bmatrix} A\epsilon_0, \dots, A^k \epsilon_0 \end{bmatrix} \tag{2.4}$$

and $\epsilon_0 = A^{-1}b - x_0$ is the initial error.

It can also be useful to think of CG in terms of polynomials:

$$x_k = x_0 + \zeta_{k-1}(A)r_0, \tag{2.5}$$

where $\zeta_{k-1}(A)$ is a polynomial of degree at most $k-1$. Then the error $\epsilon_k = A^{-1}b - x_k$ and the residual $r_k = b - Ax_k$ satisfy

$$\epsilon_k = \rho_k(A)\epsilon_0, \qquad r_k = \rho_k(A)r_0, \qquad \rho_k(A) = I - A\zeta_{k-1}(A). \tag{2.6}$$

Below we outline some properties of the polynomial ρ_k to be used in later sections. Let Π_k denote the set of polynomials of degree at most k. It is well known that

$$\|\epsilon_k\|_A = \|\rho_k(A)\epsilon_0\|_A = \min_{\substack{\rho\in\Pi_k \\ \rho(0)=1}} \|\rho(A)\epsilon_0\|_A. \tag{2.7}$$

Let $\mu_k^{(j)}$, $j = 1, \ldots, k$, denote the eigenvalues of T_k, known as Ritz values. These are the roots of ρ_k (see e.g. [4, §2.4], [25, §2]) which can therefore be written in the form

$$\rho_k(\lambda) = \prod_{j=1}^{k} \left(1 - \frac{\lambda}{\mu_k^{(j)}}\right). \tag{2.8}$$

In the following lemma we give another characterization of ρ_k. Similar ideas are used in [7, §1] and [18, §6].

LEMMA 2.1. *Let the spectral decomposition of A be*

$$A = Q\Lambda Q^T, \quad Q^{-1} = Q^T, \quad \Lambda = \mathrm{diag}(\lambda_i), \quad 0 < \lambda_1 \leq \cdots \leq \lambda_n,$$

and define

$$L_k = \begin{bmatrix} \lambda_1 & \ldots & \lambda_1^k \\ \vdots & & \vdots \\ \lambda_n & \ldots & \lambda_n^k \end{bmatrix}, \quad w = \Lambda^{1/2}Q^T\epsilon_0, \quad W = \mathrm{diag}(w_i), \quad e = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}. \tag{2.9}$$

Write the polynomial ρ_k in (2.6) *as*

$$\rho_k(\lambda) = 1 + \sum_{i=1}^{k} \tau_i \lambda^i, \qquad t_k = [\tau_1, \ldots, \tau_k]^T. \tag{2.10}$$

Then provided CG has not converged to the exact solution $A^{-1}b$ before step k, the matrix $L_k^T W^2 L_k$ is non-singular and the vector of coefficients t_k satisfies

$$t_k = -(L_k^T W^2 L_k)^{-1} L_k^T W^2 e. \tag{2.11}$$

Proof. If the matrix $L_k^T W^2 L_k$ is singular there exists a nonzero vector $v \in \Re^k$ such that $WL_kv = 0$. This implies, for all indices $i \in [1, k]$, either $w_i = 0$ or

$$v_1\lambda_i + v_2\lambda_i^2 + \cdots + v_k\lambda_i^k = 0.$$

Let j denote the smallest index such that $v_j \neq 0$, and without loss of generality scale v such that $v_j = 1$. Then either $w_i = 0$ or

$$1 + v_{j+1}\lambda_i + \cdots + v_k\lambda_i^{k-j} = 0.$$

In other words, there is a polynomial $\tilde{\rho}_{k-j}$ of degree at most $k-j$ such that $\tilde{\rho}_{k-j}(0) = 1$ and $\tilde{\rho}_{k-j}(\lambda_i) = 0$ for all i satisfying $w_i \neq 0$. Note from the definition of w in (2.9) that $w_i = 0$ implies $e_i^T Q\epsilon_0 = 0$. Thus, we have

$$\tilde{\rho}_{k-j}(A)\epsilon_0 = Q\tilde{\rho}_{k-j}(\Lambda)Q^T\epsilon_0 = 0.$$

The above and (2.7) implies that $\epsilon_{k-j} = 0$, i.e., $x_{k-j} = A^{-1}b$. Therefore, so long as CG has not converged to the exact solution, the matrix $L_k^T W^2 L_k$ cannot be singular.

Now from (2.7),

$$\|\rho_k(A)\epsilon_0\|_A = \min_{\substack{\rho\in\Pi_k \\ \rho(0)=1}} \|\rho(\Lambda)w\|_2^2 = \min_{\substack{\rho\in\Pi_k \\ \rho(0)=1}} \|W\rho(\Lambda)e\|_2^2 = \min_t \|W(e + L_k t)\|_2^2.$$

In other words, the vector of coefficients t_k in (2.10) is the solution of a weighted linear least-squares problem:

$$t_k = \arg\min_t \|W(e + L_k t)\|_2^2 = -(L_k^T W^2 L_k)^{-1} L_k^T W^2 e. \qquad \square$$

Lemma 2.1 gives a convenient expression for the coefficients of the polynomial ρ_k in (2.6) in terms of the eigenvalues and eigenvectors of A and the initial error ϵ_0. We use this expression to obtain our main result in the next section.

3. The Jacobian of x_k. Let x_k denote the k–th iterate of CG. The following theorem gives an expression for J_k, the Jacobian of x_k with respect to the right-hand side vector b, in terms of the matrices V_k and T_k and the polynomial ρ_k defined in Section 2. Throughout we assume that CG has not converged to the exact solution $A^{-1}b$ at step k, otherwise trivially $J_k = A^{-1}$.

THEOREM 3.1. *Let x_k be the k–th iterate of CG applied to $Ax = b$ starting from x_0. In the notation of Section 2,*

$$J_k = A^{-1}\big[I - \rho_k(A)\big] + 2V_k T_k^{-1} V_k^T \rho_k(A). \tag{3.1}$$

Proof. Because $x_k = A^{-1}b - \epsilon_k$, we have $\partial x_k = A^{-1}\partial b - \partial\epsilon_k$, where

$$\epsilon_k = \rho_k(A)\epsilon_0 = \left(I + \sum_{i=1}^k \tau_i A^i\right)\epsilon_0,$$

$$\partial\epsilon_k = \left(\sum_{i=1}^k \partial\tau_i A^i\right)\epsilon_0 + \rho_k(A)\partial\epsilon_0 = K_k\partial t_k + \rho_k(A)A^{-1}\partial b,$$

and K_k is defined in (2.4). Thus

$$\partial x_k = A^{-1}\big[I - \rho_k(A)\big]\partial b - K_k\partial t_k. \tag{3.2}$$

We now obtain an expression for the differential ∂t_k. Recall from Lemma 2.1 that

$$t_k = -(L_k^T W^2 L_k)^{-1} L_k^T W^2 e, \tag{3.3}$$

where only W in the right-hand side depends on b. Therefore, the differential of t_k is

$$\begin{aligned}\partial t_k &= (L_k^T W^2 L_k)^{-1} L_k^T 2W(\partial W) L_k (L_k^T W^2 L_k)^{-1} L_k^T W^2 e - (L_k^T W^2 L_k)^{-1} L_k^T 2W(\partial W) e \\ &= -2(L_k^T W^2 L_k)^{-1} L_k^T W(\partial W)(L_k t_k + e) \\ &= -2(L_k^T W^2 L_k)^{-1} L_k^T W \mathrm{diag}([L_k t_k + e]_i)\partial w \\ &= -2(WL_k)^\dagger \rho_k(\Lambda)\Lambda^{-1/2} Q^T \partial b,\end{aligned}$$

where we have used the facts that $\partial(M^{-1}) = -M^{-1}(\partial M)M^{-1}$ for any nonsingular matrix M and

$$\partial w = \Lambda^{1/2} Q^T \partial \epsilon_0 = \Lambda^{-1/2} Q^T \partial b,$$

which follows from the definition (2.9). Now because

$$K_k = \left[A\epsilon_0, \ldots, A^k \epsilon_0\right] = Q\Lambda^{-1/2}\left[\Lambda w, \Lambda^2 w, \ldots \Lambda^k w\right] = Q\Lambda^{-1/2} W L_k \tag{3.4}$$

we have

$$K_k \partial t_k = -2Q\Lambda^{-1/2}(WL_k)(WL_k)^\dagger \rho_k(\Lambda)\Lambda^{-1/2} Q^T \partial b. \tag{3.5}$$

Using (3.4), $\mathrm{Range}(V_k) = \mathrm{Range}(K_k)$, and $V_k^T A V_k = T_k$, we obtain

$$(WL_k)(WL_k)^\dagger = \Lambda^{1/2} Q^T K_k (K_k^T A K_k)^{-1} K_k^T Q \Lambda^{1/2} = \Lambda^{1/2} Q^T V_k T_k^{-1} V_k^T Q \Lambda^{1/2}.$$

Equation (3.1) then follows by combining the above and (3.5) into (3.2). ∎

The formula for J_k given in Theorem (3.1) involves the CG polynomial $\rho_k(A)$. Using (2.6) and (2.2) we can easily obtain equivalent expressions involving the polynomial $\zeta_{k-1}(A)$:

$$\begin{aligned} J_k &= \zeta_{k-1}(A) + 2V_k T_k^{-1} V_k^T \rho_k(A) && \text{(3.6a)} \\ &= 2V_k T_k^{-1} V_k^T + (I - 2V_k T_k^{-1} V_k^T A)\zeta_{k-1}(A) && \text{(3.6b)} \\ &= 2V_k T_k^{-1} V_k^T + (I - 2V_k V_k^T - 2\beta_{k+1} V_k T_k^{-1} e_k v_{k+1}^T)\zeta_{k-1}(A). && \text{(3.6c)} \end{aligned}$$

Another interesting remark is that J_k is generally not symmetric, since the product of the two symmetric matrices $V_k T_k^{-1} V_k^T$ and $\rho_k(A)$ in (3.1) is generally not symmetric.

If CG has converged to the exact solution at step k, i.e., if $x_k = A^{-1}b$, then $J_k = A^{-1}$. The following corollary gives an upper bound on the normwise relative error between J_k and A^{-1}.

Corollary 3.2. *In the notation of Theorem 3.1,*

$$\frac{\|J_k - A^{-1}\|_2}{\|A^{-1}\|_2} \leq 3\|\rho_k(A)\|_2. \tag{3.7}$$

Furthermore, if $\|\rho_k(A)\|_2 < 1$,

$$\frac{|r_k^T J_k r_k|}{1 + \|\rho_k(A)\|_2} \leq \|\epsilon_k\|_A^2 \leq \frac{|r_k^T J_k r_k|}{1 - \|\rho_k(A)\|_2}. \tag{3.8}$$

Proof. From (3.1) we have

$$J_k - A^{-1} = \left(-A^{-1} + 2V_k T_k^{-1} V_k^T\right)\rho_k(A).$$

The relationship (3.7) follows by takings norms and using the fact that $\|V_k T_k^{-1} V_k^T\|_2 = \|T_k^{-1}\|_2 \leq \|A^{-1}\|_2$. Recall from Section 2 that $r_k^T V_k = 0$. Therefore,

$$\|\epsilon_k\|_A^2 = r_k^T A^{-1} r_k = r_k^T \big(J_k + A^{-1}\rho_k(A)\big) r_k = r_k^T J_k r_k + r_k^T A^{-1/2} \rho_k(A) A^{-1/2} r_k,$$

from which the inequalities in (3.8) follow. ∎

We can interpret Corollary 3.2 as follows. If $\|\rho_k(A)\|_2 \ll 1$ then the Jacobian J_k is close to A^{-1} and the energy norm of the error is close to $(r_k^T J_k r_k)^{1/2}$. From the characterization of ρ_k in (2.7),

$$\|\rho_k(A)\|_2 \geq \frac{\|\rho_k(A)\epsilon_0\|}{\|\epsilon_0\|} = \frac{\|\epsilon_k\|}{\|\epsilon_0\|},$$

where $\|\cdot\|$ can denote either the A-norm, the 2-norm, or the A^2-norm, i.e., the residual 2-norm. If ϵ_0 has a significant component along the eigenvector of A corresponding to the largest in magnitude eigenvalue of $\rho_k(A)$, the above lower bound is a reasonable approximation. If this is the case, and if $\|\epsilon_k\|/\|\epsilon_0\| \ll 1$, then $\|\rho_k(A)\|_2 \ll 1$. However, it is certainly possible to construct examples in which $\|\rho_k(A)\|_2 \gg 1$. One such example is given in Section 5.

We can also use Theorem 3.1 to obtain bounds on $\|J_k\|_2$, the spectral norm condition number of x_k with respect to perturbations in b.

COROLLARY 3.3. *In the notation of Theorem 3.1, with the polynomial* ζ_{k-1} *defined in* (2.5),

$$\|T_k^{-1} e_1\|_2 \leq \|J_k\|_2 \leq 2\|T_k^{-1}\|_2 + \big(1 + 2\|A\|_2 \|T_k^{-1}\|_2\big)\|\zeta_{k-1}(A)\|_2. \tag{3.9}$$

Proof. For the lower bound, from (3.6a) we have

$$\|J_k\|_2 \geq \frac{\|J_k r_0\|_2}{\|r_0\|_2} = \frac{\big\|\zeta_{k-1}(A) r_0 + 2V_k T_k^{-1} V_k^T \rho_k(A) r_0\big\|_2}{\|r_0\|_2}.$$

Recall that $V_k^T \rho_k(A) r_0 = V_k^T r_k = 0$. Furthermore, using (2.5), (2.3), and the fact that $\beta_1 = \|r_0\|_2$ we have

$$\|\zeta_{k-1}(A)\|_2 \geq \frac{\|\zeta_{k-1}(A) r_0\|_2}{\|r_0\|_2} = \|T_k^{-1} e_1\|_2. \tag{3.10}$$

The upper bound is an immediate consequence of (3.6b). ∎

If r_0 has a significant component along the eigenvector of A corresponding to the largest eigenvalue of $\zeta_{k-1}(A)$, then the lower bound in (3.10) is a reasonable approximation and $\|\zeta_{k-1}(A)\|_2 \approx \|T_k^{-1} e_1\|_2$. If this is the case, (3.9) shows that both the lower bound and upper bound on the spectral norm of the Jacobian depend only on terms involving T_k^{-1} (as opposed to A^{-1}). In such cases, the spectral norm condition number of x_k is essentially determined by the reciprocal of the smallest Ritz value. Of course, this reasoning does not hold in the worst case, and it may happen that $\|J_k\|_2 \gg \|T_k^{-1}\|_2$. Nevertheless, in typical problems, the 2-norm condition number of CG iterates is often determined by the smallest Ritz value. Numerical examples are given in Section 5.

4. Computing Matrix-Vector Products. In most practical applications, due to memory limitations it is not possible to explicitly compute and store the entire matrix $J_k \in \Re^{n\times n}$. In this section we show how matrix-vector products (matvecs) $J_k v$ and $J_k^T v$ can be computed or estimated. This can be used to estimate the spectral norm and Frobenius norm condition numbers of x_k, as follows: for any v of unit length we have a lower bound $\|J_k\|_2 \geq \|J_k v\|_2$, while if $v \sim (0, I)$ then $\|J_k v\|_2^2$ is an unbiased estimator of $\|J_k\|_F^2$.

4.1. Formulas involving polynomials. We can use the expressions in (3.6) to compute the matvecs $J_k v$ and/or $J_k^T v$. There are several mathematically (but not numerically) equivalent methods of doing this. The computation of a degree-k polynomial in A times a vector using Horner's method involves $k-1$ matvecs with A. This is essentially as expensive as performing $k-1$ extra iterations of CG. Other algorithms for computing $\rho_k(A)v$ involve as few as $\sqrt{k}$ matvecs with A, but require at least n^2 storage; see, e.g., [13, §5]. This is much likely too expensive to be practical in large sparse applications. Even if we are willing to perform k extra matvecs with A, the accurate computation of $\rho_k(A)v$ and/or $\zeta_{k-1}(A)v$ in floating point arithmetic is very challenging, particularly for larger values of k. (See below.)

Nevertheless, the formulas in (3.6) can be used as a starting point to obtain cheap estimates of $J_k v$ and $J_k^T v$. In the following estimates the matrices V_k and T_k are explicitly used. Therefore, these estimates really apply to the Lanczos algorithm for solving symmetric positive definite $Ax = b$ (which is mathematically equivalent to CG but in which one explicitly computes and stores V_k and T_k).

A natural idea is to estimate J_k by replacing A with an estimate of A in (3.6). Using $A \approx V_k T_k V_k^T$ we obtain

$$\begin{aligned} J_k &= \zeta_{k-1}(A) + 2V_k T_k^{-1} V_k^T \rho_k(A) \\ &\approx \zeta_{k-1}(V_k T_k V_k^T) + 2V_k T_k^{-1} V_k^T \rho_k(V_k T_k V_k^T) \\ &= V_k \zeta_{k-1}(T_k) V_k^T + 2V_k T_k^{-1} \rho_k(T_k) V_k^T \\ &= V_k T_k^{-1} \big[I - \rho_k(T_k)\big] V_k^T + 2V_k T_k^{-1} \rho_k(T_k) V_k^T = V_k T_k^{-1} V_k^T. \end{aligned}$$

The last equality follows from the fact that $\rho_k(T_k) = 0$, since the eigenvalues of T_k are the zeros of ρ_k. If $A \approx V_k T_k V_k^T$ is a good approximation then hopefully so is $J_k \approx V_k T_k^{-1} V_k^T$, but this depends on the sensitivity of the polynomials $\zeta_{k-1}(A)$ and $\rho_k(A)$ to perturbations in A. See for example [15, §6.1] for one such perturbation result. Also note that if $v \perp \text{Range}(V_k)$, clearly $V_k T_k^{-1} V_k^T v = 0$ may be very far from $J_k v$.

Another simple idea is to replace $\rho_k(A)$ and/or $\zeta_{k-1}(A)$ in (3.6) with lower-degree polynomials. For example, from (3.6c) we can approximate

$$J_k v \approx f_{k,d^*} = 2V_k T_k^{-1} V_k^T v + (I - 2V_k V_k^T - 2\beta_{k+1} V_k T_k^{-1} e_k v_{k+1}^T)\zeta_{d-1}(A)v, \quad (4.1)$$

where $d = \min\{k, d^*\}$ for some small index d^*. At step $k = d^*$, we compute the vector $\zeta_{d^*-1}(A)v$, store it, and reuse it for the following iterations. Computing the resulting approximation to $J_k v$ for all $k > d^*$ no longer requires any matvecs with A. Unfortunately, we have not found a way to exploit this idea to compute $J_k^T v$, since in this case the vector

$$(I - 2V_k V_k^T - 2\beta_{k+1} v_{k+1} e_k^T T_k^{-1} V_k^T)v$$

to be multiplied by $\zeta_{d-1}(A)$ changes at every iteration.

It is known that the polynomial ζ_{k-1} follows a three-term recurrence relation (see [19, Lemma 2.2.5]). For completeness we state it here in our notation. Recall that the scalars α_k and β_k are the entries of T_k in (2.1). Denote

$$\eta_k = \frac{\alpha_k}{\beta_{k+1}} \frac{\|r_k\|_2}{\|r_{k-1}\|_2}.$$

Then

$$\begin{aligned}
&\zeta_0(\lambda) = \alpha_1^{-1}, \\
&\zeta_1(\lambda) = \eta_2(\alpha_1^{-1} + \alpha_2^{-1} - \alpha_1^{-1}\alpha_2^{-1}\lambda), \\
&\zeta_{k-1}(\lambda) = \alpha_k^{-1}\eta_k + \eta_k(1 - \alpha_k^{-1}\lambda)\zeta_{k-2}(\lambda) + (1 - \eta_k)\zeta_{k-3}(\lambda), \quad k = 3, 4, \ldots.
\end{aligned}$$

One can also derive an equivalent relation that uses the CG coefficients μ_k and ν_k (in Algorithm 1 below) instead of the Lanczos coefficients α_k and β_k (see [19, §2.6]) but because (3.6c) and (4.1) require V_k and T_k from the Lanczos algorithm we consider the recurrence as stated above. This immediately gives a three-term recurrence for computing the vectors $\zeta_{k-1}(A)v$. Each step in the recurrence requires one matvec with A. This can be done on the fly.

It is also possible to compute $\zeta_{k-1}(A)v$ from its coefficients in the monomial basis. We first compute the Ritz values, which are the roots of ρ_k. We can then interpolate at the points $\{(0,1), (\mu_k^{(1)}, 0), \ldots, (\mu_k^{(k)}, 0)\}$ to obtain the coefficients of ρ_k. From this and (2.6) we immediately obtain the coefficients of ζ_{k-1}, from which we can compute $\zeta_{k-1}(A)v$ using Horner's method.

4.2. Automatic differentiation. For a given vector v, the matvecs $J_k v$ and $J_k^T v$ can also be computed directly using automatic differentiation techniques. (See for example [22, §7.2] for an introduction to automatic differentiation.) In the following we outline how to do this.

Algorithm 1 The standard CG iterations

Given A, b, and x_0
$r_0 = b - Ax_0$
$p_0 = r_0$
$k = 1$
while stopping criterion not satisfied **do**
  $\mu_k = r_{k-1}^T r_{k-1} \,/\, p_{k-1}^T A p_{k-1}$
  $x_k = x_{k-1} + \mu_k p_{k-1}$
  $r_k = r_{k-1} - \mu_k A p_{k-1}$
  $\nu_k = r_k^T r_k \,/\, r_{k-1}^T r_{k-1}$
  $p_k = r_k + \nu_k p_{k-1}$
  $k = k + 1$
end while
end

Algorithm 1 defines the standard CG iterations, presented essentially as in [19]. We can recast one full iteration of CG in terms of the action of operators. First we

replace the scalars μ_k and ν_k by their corresponding expressions

$$\begin{aligned}\mu_k &= \frac{r_{k-1}^T r_{k-1}}{p_{k-1}^T A p_{k-1}},\\ \nu_k &= \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}} = \frac{(r_{k-1}+\mu_{k-1}Ap_{k-1})^T(r_{k-1}+\mu_{k-1}Ap_{k-1})}{r_{k-1}^T r_{k-1}}\\ &= 1 + 2\frac{p_{k-1}^T A r_{k-1}}{p_{k-1}^T A p_{k-1}} + \frac{r_{k-1}^T r_{k-1})(p_{k-1}^T A A p_{k-1})}{(p_{k-1}^T A p_{k-1})^2}.\end{aligned}$$

Define the vector $z_k = [x_k^T, r_k^T, p_k^T]^T \in \Re^{3n}$. For the initial step (lines 2–4) we have

$$z_0 = F_0(b) = \begin{bmatrix} x_0 \\ b - Ax_0 \\ b - Ax_0 \end{bmatrix}, \qquad \frac{\partial z_0}{\partial b} = \begin{bmatrix} 0 \\ I_n \\ I_n \end{bmatrix}.$$

At step k of CG (lines 6–8) we have

$$z_k = F\left(z_{k-1}\right) = \left[\begin{array}{l} x + \frac{r^T r}{p^T A p} p \\ r + \frac{r^T r}{p^T A p} A p \\ -r - \frac{r^T r}{p^T A p} A p + \left(1 + 2\frac{p^T A r}{p^T A p} + \frac{(r^T r)(p^T A A p)}{(p^T A p)^2}\right) p \end{array}\right]_{k-1}.$$

Differentiating the above, we obtain

$$\frac{\partial z_k}{\partial z_{k-1}} = \left[\begin{array}{lll} I_n & \frac{2}{\theta} p r^T & \mu I_n - 2\frac{\mu}{\theta} p q^T \\ 0_n & I_n + \frac{2}{\theta} q r^T & \mu A - 2\frac{\mu}{\theta} q q^T \\ 0_n & -I_n - \frac{2}{\theta} q r^T + \frac{2}{\theta} p q^T + \frac{2\tau}{\theta^2} p r^T & G \end{array}\right]_{k-1}.$$

with $q = Ap$, $\mu = \frac{r^T r}{p^T q}$, $\theta = p^T q$, $\tau = q^T q$, and

$$\begin{aligned} G = -\mu A + \frac{2\mu}{\theta} q q^T + \left(1 + \frac{2q^T r}{\theta} + \frac{\mu\tau}{\theta}\right) I_n \\ + \frac{2}{\theta} p r^T A - \frac{4 q^T r}{\theta^2} p q^T + \frac{2\mu}{\theta} p q^T A - \frac{4\mu\tau}{\theta^2} p q^T. \end{aligned}$$

Applying the chain rule to successive iterations we have

$$J_k = \frac{\partial x_k}{\partial b} = [I_n, 0, 0] \frac{\partial z_k}{\partial b} = [I_n, 0, 0] \left(\frac{\partial z_k}{\partial z_{k-1}}\right) \ldots \left(\frac{\partial z_1}{\partial z_0}\right) \left(\frac{\partial z_0}{\partial b}\right). \tag{4.2}$$

For any vector v, one can compute

$$J_k v = [I_n, 0, 0] \left(\frac{\partial z_k}{\partial z_{k-1}}\right) \ldots \left(\frac{\partial z_1}{\partial z_0}\right) \left(\frac{\partial z_0}{\partial b}\right) v \tag{4.3}$$

on the fly by updating $v = \left(\frac{\partial z_k}{\partial z_{k-1}}\right) v$ at step k of CG. The main cost of this operation is one matvec with A for each update step. This is known as the "forward" or "direct" mode in automatic differentiation.

Computing

$$J_k^T v = \left(\frac{\partial z_0}{\partial b}\right)^T \left(\frac{\partial z_1}{\partial z_0}\right)^T \ldots \left(\frac{\partial z_k}{\partial z_{k-1}}\right)^T [I_n, 0, 0]^T v,$$

cannot be done on the fly, since

$$\left(\frac{\partial z_i}{\partial z_{i-1}}\right)^T \cdots \left(\frac{\partial z_k}{\partial z_{k-1}}\right)^T [I_n, 0, 0]^T v,$$

is not know at step i of CG. One has to store (or recompute) all the quantities involved at every step in CG and loop through the algorithm in reverse order a posteriori. This is known as the "reverse" or "adjoint" mode.

In our experience, if one is willing to perform k extra matvecs with A, the most accurate method of computing matvecs with J_k is using automatic differentiation. Numerical examples are given in the next section.

5. Numerical Experiments. We provide some numerical experiments to illustrate the theory developed in the previous section.

5.1. Convergence of $\|J_k\|_2$. First we illustrate the bounds on $\|J_k\|_2$ from Section 3 by showing how the convergence of $\|J_k\|_2$ is related to that of $\|T_k^{-1}\|_2$.

Example 5.1(a) is meant to illustrate worst-case, but highly unlikely, behaviour of $\|J_k\|_2$. In this example A is a diagonal 128×128 matrix with 64 eigenvalues equally spaced in $[10^{-3}, 10^{-2}]$ and 64 eigenvalues equally spaced in $[10^2, 10^3]$. The right-hand side vector is $b = \begin{bmatrix}1, \dots 1, 0, \dots, 0\end{bmatrix}^T$ and the iteration is started with $x_0 = 0$. Because r_0 is orthogonal to all eigenvectors corresponding to the eigenvalues in $[10^2, 10^3]$, the Lanczos algorithm fails to compute any Ritz value in this interval. Consequently, for all k,

$$\begin{aligned} \|\rho_k(\Lambda)\|_2 \geq |\rho_k(\lambda_n)| &= \prod_{j=1}^{k} \left|1 - \frac{\lambda_n}{\mu_j}\right| \geq \prod_{j=1}^{k} \left(\frac{10^3}{10^{-2}} - 1\right) \approx 10^{5k}, \\ \|\zeta_{k-1}(\Lambda)\|_2 \geq |\zeta_{k-1}(\lambda_n)| &= \frac{1}{\lambda_n}|1 - \rho_k(\lambda_n)| \approx 10^{5k-3}. \end{aligned} \tag{5.1}$$

In other words, the condition number of x_k grows very quickly as k increases, and the iterates are soon highly sensitive to perturbations in b. This is illustrated in Figures 5.1 and 5.2.

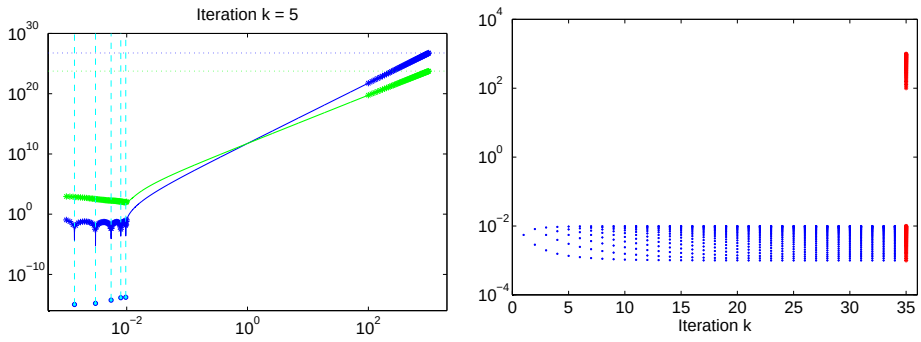

FIG. 5.1. *Left: Polynomials $\zeta_4(\lambda)$ (light) and $\rho_5(\lambda)$ (dark) plotted vs λ and in particular evaluated at the eigenvalues of A (stars). The vertical lines show the zeros of ρ_5, i.e., the Ritz values $\mu_5^{(j)}$, $j = 1, \dots, 5$. The horizontal lines denote $\|\zeta_4(\Lambda)\|_2$ and $\|\rho_5(\Lambda)\|_2$. Right: Convergence of the Ritz values as k increases. The eigenvalues of A are shown at the right. No Ritz value converges to an eigenvalue in $[10^2, 10^3]$.*

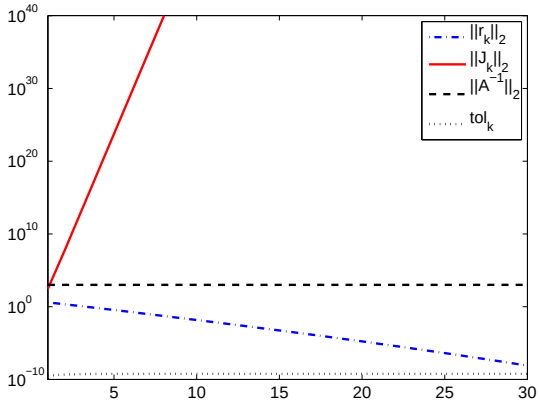

FIG. 5.2. *Convergence of* $\|J_k\|_2$ *vs* k *in CG for example 5.1(a). As predicted by* (5.1), $\|J_k\|_2$ *increases with a slope close to 5 on the semilog plot. The plotted tolerance is* $\text{tol}_k = \epsilon(\|A\|_2\|x_k\|_2 + \|b\|_2)$, *where* ϵ *is the machine unit roundoff.*

In the previous example the Ritz values corresponding to large eigenvalues failed to converge entirely. It is well established that this is highly atypical behaviour in CG. The next examples illustrate more typical behaviour. In each case the matrix A is formed via its spectral decomposition. The matrix of eigenvectors is the Q factor in the QR decomposition of a random matrix, $b = e$, and $x_0 = 0$. In example 5.1(b) the eigenvalues of A are logarithmically equally spaced between 10^{-4} and 1, while in example 5.1(c) there are $n-1$ eigenvalues of A linearly equally spaced between 1 and 10 with one extra eigenvalue 10^{-7}.

In these small examples we explicitly compute and store J_k using the automatic differentiation techniques described in Section 4. For the sake of comparison we implement CG with and without reorthogonalization (of the residual vectors). When we reorthogonalize in CG we also differentiate the reorthogonalization steps of the algorithm. To compute $\|T_k^{-1}\|_2$ we separately run the Lanczos algorithm. When we reorthogonalize in CG we also reorthogonalize in Lanczos, so that T_k is actually upper Hessenberg.

As shown in Figure 5.3, in these examples with and without reorthogonalization the spectral norm condition number essentially converges at the same rate as $\|T_k^{-1}\|_2$, which can be much smaller than $\|A^{-1}\|_2$ in the early iterations. When no reorthogonalization is used, the spectral norm of the Jacobian (as computed using automatic differentiation techniques) sometimes oscillates in the later iterations. We believe that this is a numerical artefact from the computation of J_k. The convergence of $\|J_k\|_2$ is typically smoother when reorthogonalization is used.

5.2. Computation of Matrix-Vector Products. We implement (3.6c) to compute $\|J_k v\|_2$ for a random vector $v \sim \mathcal{N}(0, I)$. We compute $\zeta_{k-1}(A)v$ both via the three-term recurrence and via interpolation of ρ_k at its roots, as described in Section 4. We also implement the related estimate $\|f_{k,d^*}\|_2$ in (4.1) with $d^* = 5$ and $d^* = 10$. Results are plotted in Figure 5.4.

Neither approach for computing $\zeta_{k-1}(A)v$ is reliable for large values of k. The accuracy of this computation seems to be problem-dependent. Formula (3.6c) gave very accurate results for test problem 5.1(c), However, for problem 5.1(b), there were severe oscillations in the computed $\|J_k v\|_2$, especially when no reorthogonalization

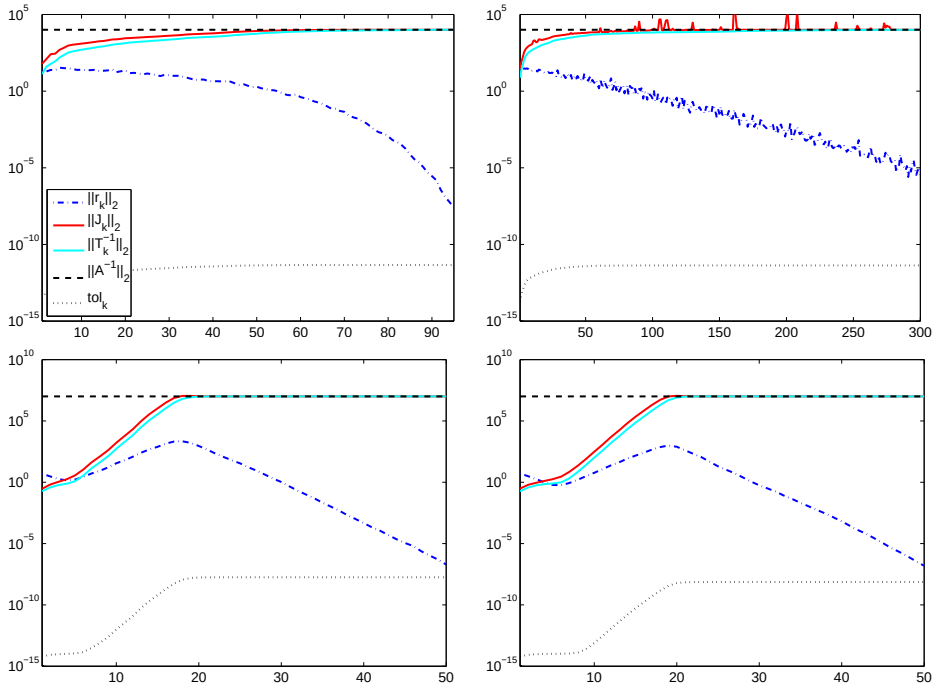

FIG. 5.3. *Convergence of $\|J_k\|_2$ vs k in CG for examples 5.1(b) and 5.1(c). Top: example 5.1(b). Bottom: example 5.1(c). Left: with reorthogonalization. Right: without reorthogonalization. The plotted tolerance is* $\mathrm{tol}_k = \epsilon(\|A\|_2\|x_k\|_2 + \|b\|_2)$, *where ϵ is the machine unit roundoff.*

was used. Even with reorthogonalization, in problem 5.1(b) the quantity $\|J_k v\|_2$ with $\zeta_{k-1}(A)v$ computed using the three-term recurrence diverged.

In (4.1), however, the quantity $\zeta_{d-1}(A)v$ is computed only for small values of $d \le d^*$. In these examples, each method for computing $\zeta_{d-1}(A)v$ gave similar results, so we only plot the estimate with $\zeta_{d-1}(A)v$ computed using the three-term recurrence. The estimate $\|f_{k,d^*}\|_2$ in (4.1) is a reasonable approximation to $\|J_k v\|_2$, even for large values of $k > d^*$. It tends to under-estimate $\|J_k v\|_2$ in the early iterations (but less so than $\|V_k T_k^{-1} V_k^T v\|_2$) and slightly over-estimate $\|J_k v\|_2$ in the later iterations.

6. Discussion. We have performed a first order perturbation analysis for CG iterates. Our results quantify how sensitive the iterates are to perturbations in the right-hand side vector, which is of interest in many applications. In Theorem 3.1 we obtained an expression for the Jacobian of x_k in CG in terms of the matrices V_k and T_k from the Lanczos algorithm and the polynomials ρ_k in (2.6). We used this result to bound the spectral norm condition number of x_k. We showed that the condition number in the spectral norm typically converges as $\|T_k^{-1}\|_2$, the reciprocal of the smallest Ritz value, and this quantity can be much smaller in the early iterations than $\|A^{-1}\|_2$. This quantifies the regularization property of CG that has been observed in practise.

We have also discussed methods to compute or estimate the matvecs $J_k v$ and $J_k^T v$. In our experience, automatic differentiation seems to be the most reliable way to compute $J_k v$ and $J_k^T v$, at the cost of k extra matvecs with A (as well as extra storage if $J_k^T v$ is required). So far we have not found a way to compute $J_k v$ or $J_k^T v$

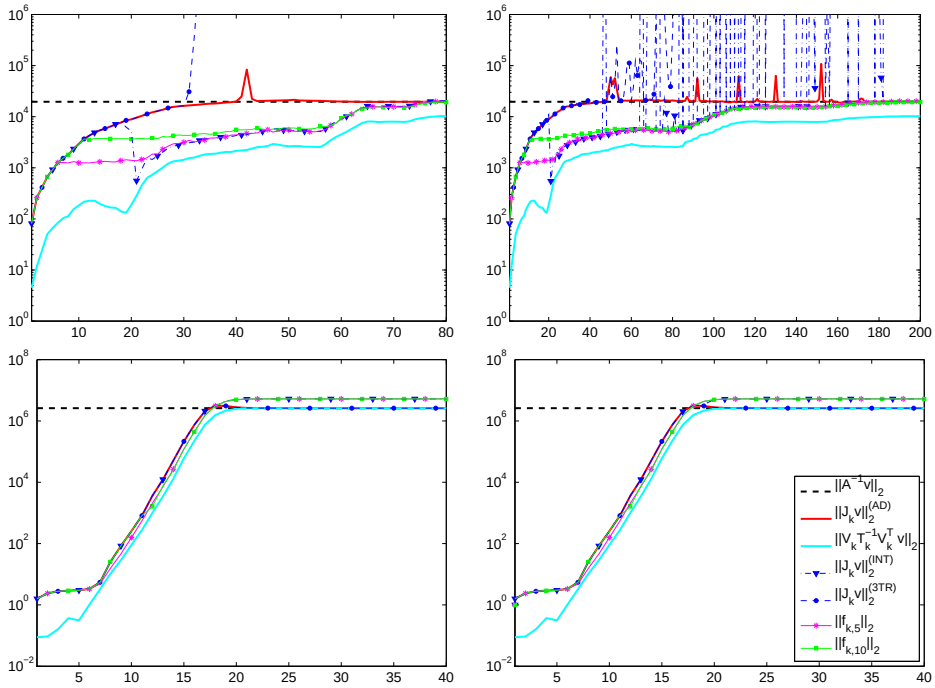

FIG. 5.4. *Convergence of $\|J_k v\|_2$ and its estimates vs k for examples 5.1(b) and 5.1(c). Top: example 5.1(b). Bottom: example 5.1(c). Left: with reorthogonalization. Right: without reorthogonalization. In the legend, AD stands for automatic differentiation, 3TR for three-term recurrence, and INT for interpolation.*

very accurately without performing k extra matvecs with A. We have shown that a cheap and easy way to estimate J_k is $J_k \approx V_k T_k^{-1} V_k^T$. We have also proposed schemes that can improve the accuracy of this estimate using fewer than k matvecs with A. Whether or not this can be improved upon is an open question.

In deriving Theorem 3.1 we have analyzed the Lanczos and CG algorithms assuming exact arithmetic. In finite precision the relationship (2.2) holds to machine precision; however, the columns of V_k can quickly lose their orthogonality and even their linear independence. Paige [23, 24] has recently showed that there exists a matrix H_k with $\|H_k\|_2 \leq \epsilon \|A\|_2$, ϵ the unit roundoff, such that the matrix T_k obtained in finite precision satisfies

$$\left(\begin{bmatrix} A & \\ & T_k \end{bmatrix} + H_k \right) Q_k = Q_k T_k + \beta_{k+1} q_{k+1} e_k^T,$$

where the columns of Q_k are exactly orthogonal and $q_1 = [v_1^T, 0]^T$. In other words, the computed T_k is the tridiagonal matrix produced by the exact Lanczos process applied to a small perturbation of an augmented matrix $\mathrm{diag}(A, T_k)$ started with the augmented vector $[v_1^T, 0]^T$. Paige called the above the augmented backward stability of the Lanczos algorithm. In the future we intend to use this result to analyze the true sensitivity of the Lanczos and CG algorithms implemented in floating point arithmetic. (By this we do not mean the difference between x_k and $\tilde{x}_k$, the iterates computed in exact and floating point arithmetic, respectively, but rather between $\tilde{x}_k(b)$ and $\tilde{x}_k(b + \delta b)$.)

We are working on extending the ideas presented in this manuscript to minimum-residual and other polynomial-based iterative methods. For CG and other methods we can also consider other derivatives such as $\frac{\partial x_k}{\partial x_0}$, $\frac{\partial \|r_k\|_2}{\partial r_0}$, etc. It may also be possible to find sparse approximations of these derivatives. Another interesting and very challenging problem which we have not considered here is the computation of the derivative of x_k with respect to elements of A.

REFERENCES

[1] J.-F. Carpraux, S. K. Godunov, and S. V. Kuznetsov. Condition number of the Krylov bases and subspaces. *Linear Algebra and its Applications*, 248(1):137–160, 1996.

[2] J. W. Demmel. *Applied numerical linear algebra*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.

[3] L. Eldén. Partial least-squares vs. Lanczos bidiagonalization—I: analysis of a projection method for multiple regression. *Computational Statistics & Data Analysis*, 46:11–31, 2004.

[4] B. Fischer. *Polynomial Based Iteration Methods for Symmetric Linear Systems*. SIAM, Philadelphia, USA, 2011.

[5] M. Fischer and P. Courtier. Estimating the covariance matrices of analysis and forecast error in variational data assimilation. Technical Report 220, European Centre for Medium-Range Weather Forecasts, Reading, UK, 1995.

[6] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, third edition, 1996.

[7] A. Greenbaum. Comparison of splittings used with the conjugate gradient algorithm. *Numerische Mathematik*, 33(1):181–194, 1979.

[8] A. Greenbaum. Behavior of slightly perturbed Lanczos and conjugate gradient recurrences. *Linear Algebra and its Applications*, 113(1):7–63, 1989.

[9] A. Greenbaum. *Iterative Methods for Solving Linear Systems*. SIAM, Philadelphia, USA, 1997.

[10] A. Greenbaum and Z. Strakoš. Predicting the behaviour of finite precision Lanczos and conjugate gradient computations. *SIAM Journal on Matrix Analysis and Applications*, 13(1):121–137, 1992.

[11] P. C. Hansen. *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1998.

[12] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of the National Bureau of Standards*, 49:409–436, 1952.

[13] N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, Philadelphia, USA, second edition, 2002.

[14] R. R. Hocking. *Methods and Applications of Linear Models: Regression and the Analysis of Variance*. John Wiley & Sons, New York, NY, USA, 2nd edition, 2003.

[15] R. A. Horn and C. R. Johnson. *Topis in Matrix Analysis*. Cambridge University Press, Cambridge, England, 1999.

[16] S. V. Kuznetsov. Perturbation bounds of the Krylov bases and associated Hessenberg forms. *Linear Algebra and its Applications*, 265(1):1–28, 1997.

[17] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of research of the National Bureau of Standards B*, 45:225–280, 1950.

[18] G. Meurant. *Computer Solution of Large Linear Systems*. North Holland, Amsterdam, 1999.

[19] G. Meurant. *The Lanczos and Conjugate Gradient Algorithms: From Theory to Finite Precision Computations*. SIAM, Philadelphia, USA, 2006.

[20] G. Meurant and Z. Strakoš. The Lanczos and conjugate gradient algorithms in finite precision arithmetic. *Acta Numerica*, 15:471–542, 2006.

[21] A. M. Moore, H. G. Arango, and G. Broquet. Estimates of analysis and forecast errors derived from the adjoing of 4D-Var. *Submitted for publication*, 2012.

[22] J. Nocedal and S. J. Wright. *Numerical Optimization*. Series in Operations Research. Springer Verlag, Heidelberg, Berlin, New York, 1999.

[23] C. C. Paige. A useful form of unitary matrix obtained from any sequence of unit 2-norm n-vectors. *SIAM Journal on Matrix Analysis and Applications*, 31(2):565–583, 2009.

[24] C. C. Paige. An augmented stability result for the lanczos hermitian matrix tridiagonalization process. *SIAM Journal on Matrix Analysis and Applications*, 31(5):2347–2359, 2010.

[25] C. C. Paige, B. Parlett, and H. van der Vorst. Approximate solutions and eigenvalue bounds from Krylov subspaces. *Numerical Linear Algebra with Applications*, 2(2):115–133, 1995.

[26] Z. Strakoš. On the real convergence rate of the conjugate gradient method. *Linear Algebra and its Applications*, 154-156:535–549, 1991.